CENTER FOR CYBERNETICS STUDIES

The University of Texas
Austin, Texas 78712

32

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
Center for Cybernetic Studies
The University of Texas

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3. REPORT TITLE
A Comparison of Computation Times for Various Starting Procedures, Basis Change Criteria, and Solution Algorithms for Distribution Problems

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)
Center for Cybernetic Studies - May, 1971,

5. AUTHOR(S) (First name, middle initial, last name)
Fred Glover D. Klingman
D. Karney A. Napier

6. REPORT DATE
May, 1971,

7a. TOTAL NO. OF PAGES
27

7b. NO. OF REFS
33

8a. CONTRACT OR GRANT NO.
NR 047-021

b. PROJECT NO.
N00014-67-A-0126-0008

c.
N00014-67-A-0126-0009

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
Center for Cybernetic Studies
Research Report CS 44

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT
This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES
Not applicable.

12. SPONSORING MILITARY ACTIVITY
Office of Naval Research (Code 434)
Washington, D.C.

13. ABSTRACT

New methods for accelerating the determination of basis trees and dual evaluators for distribution problems are compared with standard solution procedures in a computational study of a wide range of distribution problems of varying sizes and densities. Computer programs utilizing the new methods are tested for computational efficiency in an experiment involving four solution techniques, four start algorithms, and four change of basis criteria, thus affording an empirical determination not only of the merits of various procedures in isolation but also of their effectiveness in combination.

The study discloses that the most efficient solution procedure arises by coupling a primal transportation algorithm (embodying the accelerated updating and pricing methods) with a version of the Row Minimum start rule and a "modified first negative evaluator" rule. The resulting method was found to improve upon the efficiency of general purpose algorithms (taken from standard computer packages) by a factor of 50 or better, and also improved upon a streamlined version of the SHARE out-of-kilter code by a factor of 3. The method's median solution time for solving 175 x 175 distribution problems on a CDC 6600 computer was 11.4 seconds with a range of 9 to 13 seconds.

Unclassified
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Transportation problem						
Network problem						
Transportation times						
Linear programming times						
Network times						
Transportation starting procedures						
Transportation change of basis procedures						
Linear programming						

Unclassified
Security Classification

Graduate School of Business

Research Report
CS #44

A COMPARISON OF COMPUTATION TIMES FOR VARIOUS STARTING PROCEDURES, BASIS CHANGE CRITERIA, AND SOLUTION ALGORITHMS FOR DISTRIBUTION PROBLEMS

by

Fred Glover*
D. Karney
D. Klingman
A. Napier**

May 1971

*University of Colorado, Boulder
**Continental Oil Company

This research was partly supported by a grant from the Farah Foundation and by ONR Contracts N00014-67-A-0126-0008 and N00014-67-A-0126-0009 with the Center for Cybernetic Studies, The University of Texas. Reproduction in whole or in part is permitted for any purpose of the United States Government.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director

ABSTRACT

New methods for accelerating the determination of basis trees and dual evaluators for distribution problems are compared with standard solution procedures in a computational study of a wide range of distribution problems of varying sizes and densities. Computer programs utilizing the new methods are tested for computational efficiency in an experiment involving four solution techniques, four start algorithms, and four change of basis criteria, thus affording an empirical determination not only of the merits of various procedures in isolation but also of their effectiveness in combination.

The study discloses that the most efficient solution procedure arises by coupling a primal transportation algorithm (embodying the accelerated updating and pricing methods) with a version of the Row Minimum start rule and a "modified first negative evaluator" rule. The resulting method was found to improve upon the efficiency of general purpose algorithms (taken from standard computer packages) by a factor of 50 or better, and also improved upon a streamlined version of the SHARE out-of-kilter code by a factor of 3. The method's median solution time for solving 175 x 175 distribution problems on a CDC 6600 computer was 11.4 seconds with a range of 9 to 13 seconds.

1.0 Introduction and Scope of the Computational Analysis

The major purpose of this paper is to examine the following notions that have become part of the folklore surrounding the distribution (transportation) problem.[1]

(1) The most efficient algorithm for solving network problems is the out-of-kilter network algorithm developed by Ford and Fulkerson [11];

(2) The computational efficiency provided by special purpose algorithms for distribution and network problems is not significantly greater than that provided by general purpose linear programming methods that use sophisticated procedures for exploiting sparse matrices;

(3) The most efficient criterion (from the standpoint of total computational time) for determining a basis exchange in the context of the distribution problem is the most negative evaluator criterion [9, 20, 21];

(4) The total solution time required to solve distribution problems of dimension m x m increases proportionately to m^2 [10].

A secondary purpose of this study is to compare solution times between special purpose primal and dual simplex algorithms for solving distribution problems. Such a comparison is motivated by the conjecture [18, 19] that the special purpose dual algorithms are likely to be "faster" than their counterpart primal algorithms. The reason for this conjecture is two-fold: (a) To apply the most negative evaluator criterion to a dense m x m distribution problem in the dual approach requires the examination of only (2m-1) evaluators; whereas the primal approach requires the examination of $m^2 - (2m-1)$ evaluators; (b) there exist procedures for finding basic feasible solutions for the dual of the distribution problem while counterpart procedures (which insure that the starting basis will always be feasible) do not exist for the primal problem. Another purpose of our study, related to the foregoing, is to examine the effect on total computational time of different procedures for obtaining initial starting bases for both the primal and dual formulations of the distribution problem. The final purpose of our study is to examine the effect of the density on the solution time of an m x m distribution problem.

To guarantee a valid and impartial comparison of the procedures under analysis, the distribution problems used in the study varied between 9 percent and 100 percent density and varied in size from 10 x 10 to 200 x 200 (origins x destinations). A total of 140 different distribution problems were examined, all of which were randomly generated using a uniform probability function. The total supply of each m x m distribution problem was set equal to 1000m. The only other restrictions placed on the problems consisted of requiring the number of variables to be less than or equal to 10,000 and requiring the cost coefficients to lie between 1 and 100.

To accomplish these purposes, it has been necessary to obtain (and in some instances to develop) computer codes that are representative of the "state of the art" in solving network and linear programming problems. A description of the codes used in this study (e.g., out-of-kilter algorithm, general linear programming algorithm, and special purpose primal and dual simplex algorithms for solving distribution problems) is contained in a subsequent section.

The CDC 6600 at The University of Texas at Austin Computation Center was used to solve the distribution problems for the out-of-kilter and special purpose primal and dual simplex computer codes. Furthermore, the computer jobs were executed during periods when the machine load was approximately the same. The CDC 6600 at the CDC Data Center in Houston was used to solve the distribution problems for the general purpose simplex computer code, OPHELIE/LP.

The practical importance of determining the efficiency of alternative ways for solving distribution and network problems is affirmed not only by the fact that a sizeable fraction of the linear programming literature has been devoted to it, but also by the fact that an even larger share of the many concrete industrial and military applications of linear programming deal with distribution problems. Distribution problems often occur as subproblems in a larger problem (e.g., the traveling salesman problem or the warehouse location problem). Moreover, industrial applications of distribution problems often contain thousands of variables, and hence a streamlined algorithm is not only computationally worthwhile but a practical necessity. In addition, a number of linear programs that appear at first glance to be unrelated to the distribution problem can nevertheless be given a distribution problem formulation [7, 9, 11, 22, 23, 33], and it is also possible to approximate certain additional linear programming problems by such a formulation.

2.0 Description of Procedures Under Analysis

2.1 Solution Algorithms

The general simplex linear programming computer code employed in the study was Control Data's OPHELIE/LP code. OPHELIE/LP is a subsystem of the OPHELIE II Mathematical Programming System, and it fully exploits the characteristics of the CDC 6600 computer.

The optimization is performed by a primal algorithm based on the revised simplex method with the product form of the inverse. The algorithm performs the computation with a selection of multiple columns that is not fixed a priori but that can be altered dynamically to take into account statistics established in the course of iterations and the space available in central memory.

Two alternatives are permitted: one may successively introduce the best columns from among the remaining candidates, or optimize the linear sub-program formed by the basis and candidates. If cycling is detected during solution, an automatic perturbation procedure is triggered to remedy it. For further information concerning OPHELIE/LP, see [26, 27].

The other algorithms used in the study are designed to take direct advantage of the topological structure of the distribution problem. The special purpose primal simplex approach is the "Row-Column Sum Method" [7]. To enhance the computational efficiency of the algorithm, the predecessor and augmented predecessor index methods [17, 18] are employed to determine a basis exchange path and to effect the appropriate updating of the problem data.[3] (The basis for these procedures is the triple labeling procedure proposed by Ellis Johnson [23].)

Lemke's Dual Method [25] provides the foundation for the special purpose dual algorithm[3] (the "Dual Row-Column Sum Method"). The computer code for this algorithm modifies the dual method to take advantage of the predecessor and augmented predecessor index methods and the poly-ω method [7] to produce what has been called the "double pricing" dual algorithm [15]. Both the primal and dual row-column methods are "in core" computer codes. Each currently has a limit of 10,000 admissible cells (variables).

The out-of-kilter algorithm was tested both because of its reputation as a highly efficient solution method for network problems in general and also because of the sparseness of some of the distribution problems used in this study (which the out-of-kilter method is capable of exploiting to special advantage). The computer code for the out-of-kilter method employed in the analysis was written by R. J. Classen of the RAND Corporation and is distributed by SHARE [8, 30]. The out-of-kilter computer code is an "in core" algorithm and currently has a limit of 10,000 admissible arcs. The general form of the code was modified slightly by the authors to enhance its computational efficiency for application to distribution problems.

2.2 Start Algorithms

A variety of claims have been made concerning the influence of various "starting" algorithms, although documented evidence to support these claims has not been entirely satisfactory. To help remedy this situation, our study includes comparisons of several primal "start" methods that have appeared in the literature. The primal start methods examined are the Northwest Corner Rule [9, 20], Vogel Approximation Method (VAM) [7, 31], Row-Column Minimum [9], and Row-Minimum [10, 20]. These start algorithms do not exhaust all of those that have been proposed, but they are representative of the start procedures available for the primal formulation of the distribution problem.

The dual start method examined (which is of more recent vintage than the primal start methods) is described in [15, 19].

2.3 Basis Change Criteria

Four different criteria for specifying the variable to enter the basis (in the primal approach) were examined to ascertain their effect upon the total solution time: the commonly proposed "first negative evaluator" [10] and "most negative evaluator" [9] criteria, and variations of these called the "modified first negative evaluator" [10] and "modified most negative evaluator" criteria. The "first negative evaluator" scans the rows of the distribution tableau until it encounters the first element whose evaluator signals that this cell may "profitably" enter the basis. The "modified first negative evaluator" criterion scans the rows of the distribution tableau until it encounters the first row that contains a negative (i.e., profitable) evaluator, and then selects the most negative evaluator in this row.

The "modified most negative method" begins by selecting the most negative evaluator in the tableau on the initial iteration of the algorithm. At each subsequent iteration of the algorithm, a satisficing criterion is used that selects the first variable encountered which has an evaluator less than some multiple of the evaluator selected in the preceding iteration. If no such variable exists, the method selects the variable with the most negative evaluator. This criterion is highly flexible, since it can be made to select the same variable as the "first negative evaluator" criterion by using a zero multiple of the preceding evaluator and to select the same variable as the "most negative evaluator" criterion by using a large multiple of the preceding evaluator. The multiple used in our analysis was 1/3 (developed by the authors on the basis of preliminary testing).

In the primal approach it is important to consider the trade-off between the amount of time required to implement various basis change criteria and the number of iterations that result by using these criteria, since a method that results in fewer iterations may still consume greater overall time due to an increased time per iteration.

In the dual approach such a trade-off is not particularly important for dense $m \times m$ distribution problems since, instead of considering up to $m^2 - (2m-1)$ evaluators (as in the primal approach), the dual method has to contend with only $2m-1$ evaluators. Consequently, only the most negative basis change criterion (which tends to reduce the total number of iterations) was used in the special purpose dual simplex code.

3.0 Numerical Results and Their Interpretation

In this section we present the results of the computational experiment concerning folklores, start algorithms, solution algorithms and basis change criteria. The various median computation times appear in Tables 1-5. The computation times do not include the time for input and output.

3.1 Folklores

The results of the analysis indicate that the topological structure of the distribution model merits the use of special purpose linear programming algorithms to solve such problems. The data in Tables 1 and 2 indicate that the primal and dual row-column sum algorithms are, respectively, at least 20 times faster than the OPHELIE/LP general simplex linear programming computer code. In general the special purpose primal simplex method is 50 times faster. For instance, the solution for a 100 x 100 distribution is approximately 4.5 seconds using the special purpose code and 277 seconds using the OPHELIE/LP code. (Before computing the median times for the OPHELIE/LP code, a number of trial runs were made to determine the best procedure for selecting the next incoming cell. The trial results indicated that 30 cells should be considered at a time and, further, the most negative cell should be selected.)

Furthermore, the analysis indicates that the out-of-kilter algorithm is not the most efficient solution technique for solving all network problems. Even for extremely sparse distribution problems, as indicated in Tables 1 and 2, the special purpose primal algorithm, using the row-minimum start method and the modified first negative basis change criterion, was consistently 2 to 3 times faster than the out-of-kilter algorithm. Unexpectedly, as

Table 1

Solution Times (sec) for Out-of-Kilter,
OPHELIE/LP and Dual Algorithms

Problem Size	Density[2]	Out-of-Kilter[1] Solution Time	OPHELIE/LP[1] Solution Time	Dual[1] Solution Time	Start Time	No. Iterations	Total IT*	Time per Iteration
10 x 10	.35	.165	.755	.025	.008	8	.017	.002
20 x 20	.65	.617	4.012	.226	.024	37	.202	.005
30 x 30	.60	1.248		.689	.050	70	.639	.009
40 x 40	.36	1.748	39.375	.903	.058	83	.845	.010
50 x 50	.54	3.312		3.053	.106	144	2.947	.020
60 x 60	.20	3.639		3.026	.097	201	2.929	.015
70 x 70	.28	6.152		7.022	.149	293	6.873	.023
80 x 80	.31	11.903		9.829	.196	324	9.633	.030
90 x 90	.28	14.587		15.180	.224	446	14.956	.034
100 x 100	.20	12.302	276.90	14.622	.235	476	14.387	.030

[1] All times are median times with five problems per group.

[2] Median density.

*Iteration time.

Table 2

Total Solution Time Relative to Start Time
For Nondense Problems Using the Primal Algorithm[1]

		Vogel Approximation Method[2]					Row Minimum[2]					Row-Column Minimum[2]					Northwest Corner Rule[2]				
Problem Size	Density[3]	Solution Time	Start Time	No. of Iter.	Iteration Time	API*	Solution Time	Start Time	No. of Iter.	Iteration Time	API*	Solution Time	Start Time	No. of Iter.	Iteration Time	API*	Solution Time	Start Time	No. of Iter.	Iteration Time	API*
10 x 10	.35	.070	.045	5	.025	.005	.074	.018	12	.056	.005	.061	.017	8	.044	.006	.057	.014	11	.043	.004
20 x 20	.65	.254	.137	22	.117	.006	.260	.034	39	.226	.006	.282	.033	39	.249	.006	.590	.031	89	.502	.006
30 x 30	.60	.624	.258	66	.356	.005	.543	.046	99	.497	.005	.583	.051	82	.532	.006	1.033	.110	158	.923	.006
40 x 40	.36	.862	.469	67	.393	.006	.767	.071	110	.696	.006	.755	.067	110	.688	.006	1.186	.053	218	1.133	.006
50 x 50	.54	1.555	.717	140	.838	.006	1.284	095	168	1.189	.007	1.316	.077	195	1.239	.006	2.186	.151	308	2.035	.007
60 x 60	.20	1.593	.845	143	.748	.006	1.489	.092	204	1.397	.007	1.501	.084	228	1.417	.006	2.186	.069	355	2.117	.006
70 x 70	.28	2.575	1.356	158	1.219	.006	2.197	.129	289	2.068	.007	2.252	.163	249	2.089	.006	3.875	.092	558	3.783	.007
80 x 80	.31	3.650	1.651	248	1.999	.007	3.166	.176	378	2.990	.008	3.271	.171	378	3.100	.008	5.322	.113	740	5.209	.007
90 x 90	.28	4.347	2.068	286	2.279	.008	3.697	.176	480	3.521	.007	3.931	.174	480	3.757	.008	6.418	.141	825	6.277	.008
100 x 100	.20	4.940	2.495	291	2.445	.008	4.302	.180	505	4.122	.008	4.403	.175	505	4.228	.008	6.970	.146	906	6.824	.008
125 x 125	.37	8.301	3.979	450	4.322	.010	5.857	.299	570	5.558	.010	6.871	.548	583	6.323	.011	12.173	.220	1425	11.953	.008
150 x 150	.31	11.239	5.612	573	5.627	.010	8.218	.311	849	7.907	.009	9.040	.501	384	8.539	.010	14.942	.252	1663	14.690	.009
175 x 175	.19	13.435	7.165	596	6.270	.011	11.418	.377	1059	11.041	.010	12.677	.730	1089	11.947	.011	20.838	.346	2154	20.492	.010
200 x 200	.17	18.171	9.038	902	9.133	.010	14.443	.490	1222	13.952	.011	14.282	.510	1205	13.472	.011	23.698	.312	2477	23.386	.010

[1] The "modified first negative" basis change criterion was used in testing the start algorithms.

[2] All times are median times with five problems per group.

[3] Median density.

*Average per iteration.

Table 3

Total Solution Times Relative to Start Times
For Dense Problems Using the Primal Algorithm[1]

Problem Size Dense	Vogel Approximation Method[2] Solution Time	Start Time	No. of Iter.	Iteration Time	API*	Row Minimum[2] Solution Time	Start Time	No. of Iter.	Iteration Time	API*	Row-Column Minimum[2] Solution Time	Start Time	No. of Iter.	Iteration Time	API*	Northwest Corner Rule[2] Solution Time	Start Time	No. of Iter.	Iteration Time	API*
10 x 10	.112	.051	14	.061	.004	.074	.019	16	.055	.003	.109	.040	14	.069	.005	.143	.015	30	.128	.004
20 x 20	.296	.165	21	.131	.006	.211	.038	41	.173	.004	.328	.089	36	.239	.007	.613	.032	85	.581	.007
30 x 30	.571	.339	27	.232	.009	.440	.065	82	.375	.005	.670	.141	82	.529	.006	1.253	.052	157	1.210	.008
40 x 40	1.101	.556	66	.545	.008	.840	.096	100	.744	.007	1.027	.212	101	.815	.008	2.689	.079	282	2.610	.009
50 x 50	1.767	.885	99	.882	.009	1.211	.136	151	1.075	.007	1.642	.325	147	1.317	.009	4.069	.110	366	3.959	.011
60 x 60	2.570	1.217	133	1.353	.010	1.769	.179	181	1.590	.009	2.491	.405	215	2.086	.010	6.441	.143	533	6.298	.012
70 x 70	3.549	1.647	186	1.902	.010	2.706	.229	240	2.477	.010	2.922	.499	235	2.423	.010	9.467	.170	693	9.297	.013
80 x 80	4.640	2.171	209	2.469	.012	3.609	.283	348	3.326	.010	4.519	.615	327	3.904	.012	12.958	.214	801	12.744	.016
90 x 90	5.703	2.633	227	3.070	.014	4.128	.340	379	3.788	.010	5.845	.722	398	5.123	.013	17.533	.267	992	17.266	.017
100 x 100	6.569	3.257	223	3.312	.015	4.700	.402	386	4.298	.011	6.515	.880	410	5.635	.014	21.590	.320	1170	21.270	.018

[1] The "modified first negative" basis criteria was used in testing the start algorithms.

[2] All times are median times with ten problems per group.

*Average per iteration time.

Table 4

Total Solution Time Relative to Basis Change Criteria
For Nondense Problems Using the Primal Algorithm[1]

		First Negative[2]					Modified First Negative[2]					Modified Most Negative[2]					Most Negative[2]				
Problem Size	Density	Solution Time	Start Time	No. of Iter.	Iter. Time	API*	Solution Time	Start Time	No. of Iter.	Iter. Time	API*	Solution Time	Start Time	No. of Iter.	Iter. Time	API*	Solution Time	Start Time	No. of Iter.	Iter. Time	API*
10 x 10	.35	.066	.038	7	.028	.004	.070	.045	5	.025	.005	.061	.034	7	.027	.004	.063	.045	4	018	.005
20 x 20	.65	.274	.149	25	.125	.005	.254	.137	22	.117	.006	.255	.135	22	.120	.005	.327	.138	17	.189	.011
30 x 30	.60	.586	.219	73	.367	.005	.624	.258	66	.356	.005	.682	.274	51	.408	.008	.824	.295	29	.529	.018
40 x 40	.36	.913	.455	93	.458	.005	.862	.469	67	.393	.006	.936	.471	50	.465	.009	1.266	.466	36	.800	.022
50 x 50	.54	1.426	.722	137	.704	.005	1.555	.717	140	.838	.006	1.505	.651	112	.854	.008	2.544	.750	61	1.794	.029
60 x 60	.20	2.062	.920	219	1.142	.005	1.593	.845	143	.748	.006	1.792	.883	115	.909	.008	3.180	.908	82	2.272	.028
70 x 70	.28	2.578	1.261	231	1.317	.006	2.575	1.356	158	1.219	.007	2.793	1.320	156	1.473	.009	5.354	1.271	84	4.083	.049
80 x 80	.31	3.801	1.677	334	2.124	.006	3.650	1.651	248	1.999	.008	3.762	1.680	164	2.082	.013	6.424	1.544	110	4.880	.044
90 x 90	.28	4.969	2.061	400	2.908	.007	4.347	2.068	286	2.279	.008	4.668	2.056	198	2.612	.013	12.469	2.083	126	10.386	.082
100 x 100	.20	5.470	2.382	446	3.088	.007	4.940	2.495	291	2.445	.008	5.620	2.511	242	3.109	.013	13.394	2.429	187	10.965	.059
125 x 125	.37	8.902	4.012	561	4.890	.009	8.301	3.979	450	4.322	.010	9.777	4.041	226	5.736	.025	36.240	4.000	176	32.240	.183
150 x 150	.31	12.303	5.642	693	6.661	.010	11.239	5.612	573	5.627	.010	12.636	5.283	452	7.353	.016	42.462	5.682	233	36.780	.15P
175 x 175	.19	15.858	7.205	1002	8.653	.009	13.435	7.165	596	6.270	.011	16.227	7.188	510	9.039	.018	61.519	7.295	285	54.224	.190
200 x 200	.17	21.053	9.178	1265	11.875	.009	18.171	9.038	902	9.133	.010	20.516	9.471	555	11.045	.020					

[1] The VAM start method was used for determining an initial primal feasible basic solution.

[2] All times are median times with five problems per group.

[3] Median density.

*Average per Iteration.

Table 5

Total Solution Time Relative to Basis Change Criteria
For Dense Problems Using the Primal Algorithm[1]

Problem Size Dense	First Negative[2]					Modified First Negative[2]					Modified Most Negative[2]					Most Negative[2]				
	Solution Time	Start Time	No. of Iter.	Iter. Time	API*	Solution Time	Start Time	No. of Iter.	Iteration Time	API*	Solution Time	Start Time	No. of Iter.	Iteration Time	API*	Solution Time	Start Time	No. of Iter.	Iter. Time	API*
10 x 10	.092	.050	10	.042	.004	.112	.051	14	.061	.004	.111	.048	10	.050	.006	.133	.049	6	.04[illegible]	.008
20 x 20	.386	.161	37	.225	.006	.296	.165	21	.131	.006	.359	.154	23	.194	.008	.488	.162	17	.292	.017
30 x 30	.708	.333	48	.375	.008	.571	.339	27	.232	.009	.698	.320	34	.342	.011	1.022	.331	20	.673	.034
40 x 40	1.202	.558	90	.644	.007	1.101	.556	66	.545	.008	1.384	.573	57	.805	.016	2.851	.561	41	2.260	.056
50 x 50	1.919	.864	125	1.055	.008	1.767	.885	99	.882	.009	2.072	.841	70	1.201	.016	4.861	.883	47	3.896	.083
60 x 60	2.807	1.212	173	1.595	.009	2.570	1.217	133	1.353	.010	3.365	1.199	103	2.138	.020	9.806	1.204	73	8.580	.118
70 x 70	3.910	1.654	227	2.256	.010	3.549	1.647	186	1.902	.010	4.239	1.682	122	2.524	.022	15.978	1.648	91	14.271	.157
80 x 80	5.136	2.159	318	2.977	.009	4.640	2.171	208	2.469	.012	6.281	2.170	149	4.069	.027	26.988	2.160	122	24.767	.203
90 x 90	6.817	2.646	331	4.171	.013	5.703	2.633	227	3.070	.014	3.806	2.642	222	6.032	.032	33.614	2.640	121	30.916	.257
100 x 100	7.464	3.335	359	4.129	.012	6.569	3.257	223	3.312	.015	9.119	3.343	160	5.350	.036					

[1] The VAM start method was used to determine an initial primal feasible basic solution.

[2] All times are median times with ten problems per group.

*Average per Iteration.

indicated in Table 1, the out-of-kilter algorithm was more efficient than the special dual algorithm (but not the primal algorithm) as soon as m became at least 80.

To compare the change of basis criteria the same starting solution was used. Consequently, only the main loop times required comparison to isolate the most effective criterion. The data in Tables 4 and 5 indicate that the highly popular and often referenced "most negative" criterion for determining the next vector to enter the basis is not the most efficient basis change criterion. In fact, as indicated in Tables 4 and 5, for the basis change criteria examined in the experiment, the most negative evaluator criterion was consistently slower than the other basis change criteria examined. Furthermore, when m became larger than 100, the "most negative" criterion required at least three times as much computational effort as the other basis change criteria.

Interestingly, the data in Tables 1 and 4 indicate that the out-of-kilter algorithm is more efficient computationally than the primal row-column sum algorithm, when the latter employs the most negative evaluator criterion. Perhaps the folklore concerning the superiority of the out-of-kilter method was developed by comparing the out-of-kilter network algorithm to a transportation method that used the most negative basis change criterion. (The lack of techniques such as those indicated in Section 2.1 in the previously developed special purpose primal methods would undoubtedly have accentuated the apparent difference between primal and out-of-kilter methods.)

The data in Tables 2 and 3 (using the row-minimum start) support the finding of Dennis [10] that the total solution time required to solve m x m distribution problems increases proportionately to m^2. Also, given the same relative cost range for a distribution problem, the data indicate that the total solution time is approximately twice the time necessary to obtain an initial feasible basic solution for the VAM start in dense and nondense problems (see Tables 2 and 3).

3.2 Solution Algorithms, Start Methods and Basis Change Criteria

An important factor influencing total computation time (and hence computational efficiency) is the interrelationship between the start methods and basis change criteria. The relevant tradeoffs for the start algorithms involve the time required for obtaining an initial feasible basic solution and the number of iterations subsequently required to complete the solution of the problem. Similarly, the relevant tradeoffs for the basis change criteria involve the time consumed in searching for a new vector to enter the basis and the number of iterations required to obtain an optimal solution (time per iteration versus total number of iterations).

To determine the effect of combining various start procedures with various change of basis criteria, and to avoid excessive computational expense in making this determination, we tested each change of basis criterion relative to a single start method (VAM), and then tested each of the start methods relative to the change of basis criterion that was found to be best.

Ideally, had it not involved an enormous expenditure of computer time, it would have been desirable to have tested all combinations of start procedures and change of basis criteria against each other. However, our findings were sufficiently clear cut to make it a reasonably safe bet that the change of basis criterion and start procedure that were found to be best "in isolation" will also be best in combination. In particular, from Table 4 and Table 5 it is clear that the "modified first negative" basis is faster than the other criteria examined both in terms of the time per iteration (or "main loop time") and in terms of total solution time. (This result concurs with the finding of Dennis [10].)

In general, the number of iterations required to obtain the optimal solution was significantly reduced by using the "modified most negative" and "most negative" basis change criteria. As indicated in Tables 4 and 5, these basis change criteria led to a total iteration count that was consistently less than 80 percent of the iteration count associated with the "first negative" and "modified first negative" basis change criteria.

More importantly, however, the "first negative" and "modified first negative" basis change criteria are faster than the other criteria examined both in terms of the time per iteration (or "main loop time") and in terms of total time. The efficiency of the "first negative" and "modified first negative" criteria is accentuated as m increases and the density of the distribution problem approaches 100%. The data in Tables 4 and 5 indicate that the main loop time for the "first negative" and the "modified first negative" basis change criteria is consistently less than 70% of the main loop time required by the "modified most negative" and "most negative" basis change criteria.[4] Furthermore, these data indicate that the "modified first negative evaluator" criterion consistently requires 20 percent fewer iterations than the "first negative evaluator" criterion and the main loop time of the two criteria were approximately equal. Consequently, as indicated by the data in Tables 4 and 5, the "modified first negative" criterion was consistently at least 15 percent faster in terms of total computation time than the "first negative" criterion.

Another interesting fact that the study revealed about the basis change criteria concerns the number of possible nonbasic variables that are examined during each iteration. The "first negative" and "modified first negative" criteria always evaluate substantially fewer nonbasic variables than the "most negative" criterion, and as m increases the difference in the number of nonbasic variables evaluated also increases significantly. The data in Tables 4 and 5 indicate that when m equals 20 and 200, the number of nonbasic variables that must be examined by the "first negative" and "modified first negative" criteria as compared to the "most negative" criterion is only 50% and 5%, respectively. Thus the results indicate that the "modified first negative" basis change criterion provides the best compromise of the tradeoffs for both dense and nondense problems.

Having determined the "modified first negative" change of basis criterion to be significantly more effective than the others, we used this choice rule uniformly thereafter in conducting the tests of the alternative start methods. Again our results were clear cut. From Tables 2 and 3 it is apparent that the Row Minimum start method is best for both dense and nondense problems.

There are a number of interesting observations to be obtained from Tables 2 and 3. Vogel's method, while yielding a good start from the standpoint of the number of iterations required after it is completed, takes an inordinate amount of time to find an initial solution, and exhibits a large average iteration time for dense problems, indicating that the loop structure is complicated. (See Table 2.) Similarly, the loop structure for the Northwest Corner start is complicated, whereas the loop structures for the Row Minimum and Row-Column Minimum are simpler. On dense problems the solution time of VAM and Row-Column Minimum are approximately equal.

Comparing the results in Tables 2 and 3 yields several insights into the effect of density on primal codes. In general the nondense problems were solved only about 10% faster than dense problems using the Row Minimum start, and, in fact, the dense problems were solved faster for problem sizes not exceeding 50 x 50. Thus density does not appreciably affect total solution time. Average main loop time is substantially reduced by density, indicating that loop structure becomes less complicated as the density of a problem decreases. This reduction in average main loop time is partially offset by the fact that the number of iterations increases slightly as density decreases.

As indicated in Tables 1 and 2, the special purpose primal simplex algorithm was consistently faster than the special purpose dual simplex algorithm. It would be premature, however, to conclude that one should abandon the development of special purpose dual simplex algorithms for a number of reasons. For instance, this study indicates that unlike the primal approach, the dual approach is greatly affected by density. A 100 x 100 problem of 13% density was solved in 7.697 seconds while a 100 x 100 problem of 20% density was solved in 14.622 seconds. Although 7.697 seconds is still nearly twice the median solution time for 100 x 100 problems using the primal approach, it does suggest that the dual method may be an efficient solution procedure for solving sparse network problems. It is interesting to note that the out-of-kilter code was not nearly as sensitive to density, since its times on the above problems were 9.973 and 12.045 seconds, respectively, hence making the dual method appear still more attractive for sparse problems. Further support for this possibility is given by noting that the solution time of the primal approach varies proportionately to the number of nodes in the problem while the dual approach varies proportionately to the number of arcs in the problem.

The dual start method used in this study was not an advanced start procedure. Even so, the special purpose dual simplex algorithm consistently required fewer than 50% of the number of iterations necessary for the primal simplex algorithm when any start method except VAM was used. These observations indicate that substantial computational saving may be present if advanced start algorithms (such as those developed in [16]) are employed in conjunction with the special purpose dual simplex algorithm.

Although the dual algorithm appears to be computationally slower than the primal algorithm, it is particularly useful as compared to the primal algorithm for conducting certain types of postoptimality analyses. A principal motivation cited in the literature for applying the dual method to distribution problems (see, e.g., [2,3,7]) is the possibility that the problem's "supplies" and "demands" may not be permanently fixed, but rather are subject to change. In such a situation, the ability to begin from an optimal basis to a given problem and proceed via the dual method to an optimal solution for the altered problem is extremely useful. Such an approach is not available using the primal algorithm without extensive computation.

4.0 Concluding Remarks

In the context of the important class of linear programs known as distribution (or transportation) problems, the special purpose primal and dual simplex and the out-of-kilter algorithms have been shown to be consistently faster than the general simplex algorithm. The special purpose primal simplex algorithm is in turn consistently faster than the out-of-kilter algorithm and the dual simplex algorithm. Our results indicate that it is important to determine a "good" starting basis for the dual algorithm in order to obtain an advantageous tradeoff between the total number of iterations and the time required to implement the starting algorithm.

The computational efficiency of the special purpose primal and dual algorithms for solving distribution problems has been shown to be particularly sensitive to the search time required to find a new arc to enter the basis at each iteration. The dual algorithm is more sensitive in this regard than the primal algorithm. The relative slowness of the dual algorithm's main loop time suggests that further research on advanced dual "start" methods is worthwhile, since a relatively large amount of time could be spent finding an initial basic feasible solution and still permit an overall reduction in computation time by reducing the number of iterations.

On the other hand, a highly efficient dual start will not succeed in making the special purpose dual algorithm faster than the special purpose primal algorithm unless the main loop time for the dual method can be greatly reduced. An important qualification to this assertion rests on the fact that the special purpose dual algor'.hm appears to gain efficiency relative to the primal algorithm as the proportion of the number of nodes to number of admissible cells increases. This, combined with the fact that a basic feasible dual solution exists for networks (see [19]), indicates that the special purpose dual algorithm might be the best solution procedure for minimum cost network problems.[5]

In the domain of choice rules, the data indicate that the number of problem-solving iterations is substantially reduced by using the "most negative" and "modified most negative" basis change criteria. However, this reduced iteration count does not produce a correspondingly reduced computation time since the search time required by these criteria (particularly for dense problems) is substantially greater for the "first negative" and "modified first negative" criteria. In consequence, the latter criteria turn out to be more attractive from the standpoint of total computation time. The "modified first negative" criterion is generally the more efficient of the two, although for non-dense problems the "first negative" and "modified first negative" criteria appear to be comparably efficient.

The most efficient form of the special primal simplex algorithm was found to be based on the Row Minimum start algorithm and the "modified first negative" basis change criterion. The primal simplex algorithm employed in this study proved to be from 2 to 15 times faster than the other special purpose distribution algorithms that were available for examination.

In evaluating the performance of the methods tested, considerable effort was made to insure that the computer codes were comparably efficient in implementing the underlying algorithms, start methods, and basis change criteria. Moreover, the large differences found between the performance of the various procedures tested leads us to believe that these differences will continue to hold for improved computer codes (e.g., codes that take special advantage of the characteristics of the computer on which they are implemented). In fact the techniques used to time the codes suggest that the relative superiority of the special purpose algorithms may be even greater than indicated in Section 3. Specifically, the codes for the special purpose methods included a number of clocks to measure different aspects of the solution process, whereas the codes for the general simplex algorithms and the out-of-kilter algorithm only included clocks to measure the total solution time. When the clocks are removed from the special purpose computer codes, the computational efficiency of the algorithms has been found to increase by as much as 40 percent.[6]

Acknowledgements

The authors wish to acknowledge the cooperation of the staff of The University of Texas Computation Center, The University of Texas Business School Computation Center and the Control Data Corporation Data Center in Houston, Texas. The authors are expecially appreciative of the collaboration of Dr. Alberto Gomez-Rivas of Control Data Corporation for his aid in using the OPHELIE/LP system.

FOOTNOTES

[1] It is interesting to note that most of the folklore related to the computational requirements for solving distribution problems is based on computation performed 10 to 15 years ago. Furthermore, the distribution problems examined were often small, maximum dimension of 20 x 20 [6, 9], and they were solved either by hand or by first or second generation computers.

[2] This code was developed by F. Glover, D. Karney and D. Klingman.

[3] This code was developed by the authors.

[4] "Main loop time" (average iteration time) is the time required to test for optimality to determine the variable to enter the basis, to obtain the variable to leave the basis, to find the basis exchange path, and to update the basic variables.

[5] The authors are currently testing other dual start algorithms and new procedures for reducing the main loop time of the dual method. Network algorithms are also being tested.

[6] Incidental to our investigation was the finding--or rather confirmation--that one can make no valid conjecture about the computational efficiency of two implementations of the same algorithm unless the two computer codes are executed on the same machine using the same data and the same compiler. This fact was underscored in tests comparing our special purpose primal simplex computer code to another special purpose code, based on similar techniques, that was executed on a Univac 1108. The latter code ran faster on the Univac than our code ran on the CDC 6600, suggesting that our code was less efficient due to the fact that the CDC 6600 is generally a faster machine than the 1108.

Surprisingly, when the Univac 1108 computer code was executed on the CDC 6600, it ran slower than the special purpose primal computer code used in this experiment. The apparent reason for this reversal is that the Univac 1108 FORTRAN compiler produces more efficient object code.

This result stimulated us to check with Control Data Corporation about their compiler, which led us to discover that the CDC 6600 has two FORTRAN compilers. One produces an object code which executes about twice as fast as the other. This implies that our times could have been reduced by 50% if we had used the other compiler. Since the total compilation and execution time of the two compilers is equal, The University of Texas Computation Center only uses the less efficient object code compiler.

BIBLIOGRAPHY

[1] Balas, Egon. "The Dual Method for the Generalized Transportation Problem," Management Science Vol. 12, No. 7 (March, 1966), pp. 555-568.

[2] Balas, Egon and Hammer (Ivanescu), P. L. "On the Generalized Transportation Problem," Management Science Vol. 11, No. 1 (September, 1964), pp. 188-202.

[3] Balas, Egon and Hammer (Ivanescu), P. L. "On the Transportation Problem--Part I," Cahiers du Centre d'Etudes de Recherche Operationelle Vol. 4, No. 2 (1962).

[4] Balas, Egon and Hammer (Ivanescu), P. L. "On the Transportation Problem--Part II," Cahiers du Centre d'Eutdes de Recherche Operationelle Vol. 4, No. 3 (1962).

[5] Barr, R. S., Klingman, D., and Raike, W. M. "Computational Simplifications Through Topological Structure in Network and Distribution Models," Applied Mathematics for Management, No. AMM-13, Graduate School of Business, The University of Texas, December, 1968.

[6] Charnes, A. and Cooper, W. W. "The Stepping Stone Method of Explaining Linear Programming in Transportation Problems," Management Science Vol. 1, No. 1 (1954).

[7] Charnes, A. and Cooper, W. W. Management Models and Industrial Applications of Linear Programming Vols. I and II. New York: John Wiley and Sons, Inc., 1961.

[8] Clasen, R. J. "The Numerical Solution of Network Problems Using the Out-of-Kilter Algorithm," RAND Corporation Memorandum RM-5456-PR, Santa Monica, California, March, 1968.

[9] Dantzig, G. B. Linear Programming and Extensions. Princeton, N.J.: Princeton University Press, 1963.

[10] Dennis, J. B. "A High-Speed Computer Technique for the Transportation Problem," Journal, Association for Computing Machinery Vol. 8, (1958), pp. 132-153.

[11] Ford, L. and Fulkerson, D. "An Out-of-Kilter Method for Minimal Cost Flow Problems," Journal of SIAM Vol. 9, No. 1 (1961).

[12] Ford, L. and Fulkerson, D. "Solving the Transportation Problem," Management Science Vol. 3, No. 1 (1956).

[13] Ford, L. and Fulkerson, D. Flows in Networks. Princeton, N.J.: Princeton University Press, 1962.

[14] Fulkerson, D. "An Out-of-Kilter Method for Minimal Cost Flow Problems," Journal of SIAM Vol. 9, No. 1 (1961).

[15] Glover, Fred and Klingman, D. "Double-Pricing Dual and Feasible Start Algorithms for the Capacitated Transportation (Distribution) Problem," Applied Mathematics for Management, No. AMM-20, WP 70-58, The University of Texas at Austin, 1970.

[16] Glover, Fred and Klingman, D. "Higher Order Tree Dual Approximation Methods for the Distribution Problem," to appear in Management Science.

[17] Glover, Fred and Klingman, D. "Locating Stepping-Stone Paths in Distribution Problems via the Predecessor Index Method," Transportation Science Vol. 4, No. 2 (May, 1970).

[18] Glover, Fred, Klingman, D. and Karney, D. "The Augmented Predecessor Index Method for Locating Stepping Stone Paths and Assigning Dual Prices in Distribution Problems," to appear in Transportation Science Vol. 6, No. 1.

[19] Glover, Fred, Klingman, D. and Napier, A. "A One-Pass Algorithm to Determine a Dual Feasible Basic Solution for a Class of Capacitated Generalized Networks," to appear in Operations Research.

[20] Hadley, G. Linear Programming. Reading, Mass.: Addison-Wesley Publishing Company, Inc., 1962.

[21] Hillier, F. S. and Lieberman, G. J. Introduction to Operations Research. San Francisco, California: Holden-Day, Inc., 1967.

[22] Hitchcock, F. L. "The Distribution of a Product from Several Sources to Numerous Localities," Journal of Mathematics and Physics Vol. 20 (1941), pp. 224-230. (Cambridge: Massachusetts Institute of Technology.)

[23] Johnson, Ellis. "Networks and Basic Solutions," Operations Research Vol. 14, No. 4, 1966.

[24] Klingman, D. "On Some Unimodular and Boolean Mathematical Programming Problems," Applied Mathematics for Management, No. AMM-20, The University of Texas at Austin, May, 1969.

[25] Lemke, C. "The Dual Method of Solving Linear Programming Problems," Naval Research Logistics Quarterly Vol. 1, No. 1 (1954).

[26] Lourie, J. R. "Topology and Computation of the Generalized Transportation Problem," Management Science Vol. 11 (1964), pp. 177-187.

[27] "OPHELIE II: Mathematical Programming System," Control Data Corporation, Minneapolis, Minnesota, 1970.

[28] "OPHELIE/LP: Linear Programming Subsystem of OPHELIE II," Control Data Corporation, Minneapolis, Minnesota, 1970.

[29] Orchard-Hays, W. Advanced Linear Programming Computing Techniques. New York: McGraw-Hill Book Company, 1968.

[30] "Out-of-Kilter Network Routine," SHARE Distribution 3536, SHARE Distribution Agency, Hawthorne, New York, 1967.

[31] Reinfeld, N.V. and Vogel, W. R. Mathematical Programming. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1958.

[32] Srinivasan, V. and Thompson, G. L. "Benefit-Cost Analysis of Coding Techniques for the Primal Transportation Algorithm," Management Sciences Research Report No. 229, Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, December, 1970.

[33] Wagner, Harvey M. Principles of Operations Research. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1969.